wCorr Formulas

Paul Bailey, Ahmad Emad, Ting Zhang, Qingshu Xie

2016-03-31

The wCorr package can be used to calculate unweighted or weighted correlations of the Pearson, Spearman, polyserial, and polychoric types. By extension, the package also impelments the tetrachoric correlation as a specific case of the polychoric correlation and biserial as a specific case of the polyserial correlation. When weights are used the correlation coefficients are calculated with so called sample weights or inverse probability weights.[1]

This vignette introduces the methodology used in the wCorr package for computing the Pearson, Spearman, polyserial, and polychoric correlations, with and without weights applied. For the polyserial and polychoric correlations, the coefficient is estimated using a numerical likelihood maximization. The weighted (and unweighted) likelihood functions that are used are described.

For the polyserial and polychoric correlations, evidence is offered for the correctness of the methods, including an examination of the bias and consistency of both methods. This is done seperately for unweighted and weighted correlations. For this exercise, numerical simulations are used to show:

- the bias of the methods as a function of the true correlation coefficient (ρ) and the number of observations (n) in the unweighted and weighted cases; and
- the accuracy [measured with root mean squared error (RMSE) and mean absolute deviation (MAD)] of the methods as a function of ρ and n in the unweighted and weighed cases.

Note that here "bias" is used to mean the mean difference between true correlation and estimated correlation.

The *wCorr Arguments* vignette describes the effects the `ML` and `fast` arguments have on computation and gives examples of calls to wCorr.

Methodology

Here we focus on measurement of the correlation coefficients between two vectors of random variables that are jointly bivariate normal. We call the two vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$.[2] The i^{th} members of the vectors are then called x_i and y_i.

Methodology for Pearson and Spearman correlations

The weighted Pearson correlation is computed using the formula

$$\rho_{Pearson} = \frac{\sum_{i=1}^{n}\left[w_i(x_i-\bar{x})(y_i-\bar{y})\right]}{\sqrt{\sum_{i=1}^{n}\left(w_i(x-\bar{x})^2\right)\sum_{i=1}^{n}\left(w_i(y-\bar{y})^2\right)}}$$

where w_i is the weights, $\bar{x}$ is the weighted mean of the $\boldsymbol{X}$ variable ($\bar{x} = \frac{1}{\sum w_i}\sum_i w_i x_i$), $\bar{y}$ is the weighted mean of the $\boldsymbol{Y}$ varaible ($\bar{y} = \frac{1}{\sum w_i}\sum_i w_i y_i$), and n is the number of elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$.[3]

[1] Sample weights are comperable to `pweight` in Stata.

[2] The Spearman correlation coefficient can be derived by assuming a much more broad class of random variables, but that is not dealt with in this document.

[3] See the "correlate" function in Stata Corp, Stata Statistical Software: Release 8. College Station, TX: Stata Corp LP, 2003.

The unweighted Pearson correlation is calculated by setting all of the weights to one.

For the Spearman correlation coefficient the unweighted coefficient is calculated by ranking the data and then using those ranks to calculate the Pearson correlation coefficient–so the ranks stand in for the x and y data. Again, similar to the Pearson, for the unweighted case the weights are all set to one.

There is no commonly accepted weighted Spearman correlation coefficient. Nevertheless, we implement one possible weighted Spearman correlation coefficient as follows. First, the rank scores of the data are calculated (this ranking is unweighted) and then the ranks are passed to the Pearson correlation coefficient formula shown above, and this Pearson correlation coefficient does use the weights.

For both cases the same ranking formula is used so that the highest value recieves a value of 1 and the second highest 2, and so on down to the nth value. In addition, when data are ranked, ties must be handled in some way. The chosen method is to assign the average of all tied ranks. For example, if the second and third rank units are tied then both units would receive a rank of 2.5 (the average of 2 and 3).

Methodology for polyserial correlation with and without weights

For the polyserial correlation, it is again assumed that there are two continuous variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ that have a bivariate normal distribution.[4]

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \Sigma\right]$$

Where $N(A, \Sigma)$ is a bivariate normal distribution with mean vector A and covariance matrix Σ. For the polyserial correlation, $\boldsymbol{Y}$ is discretized into the random variable $\boldsymbol{M}$ according to

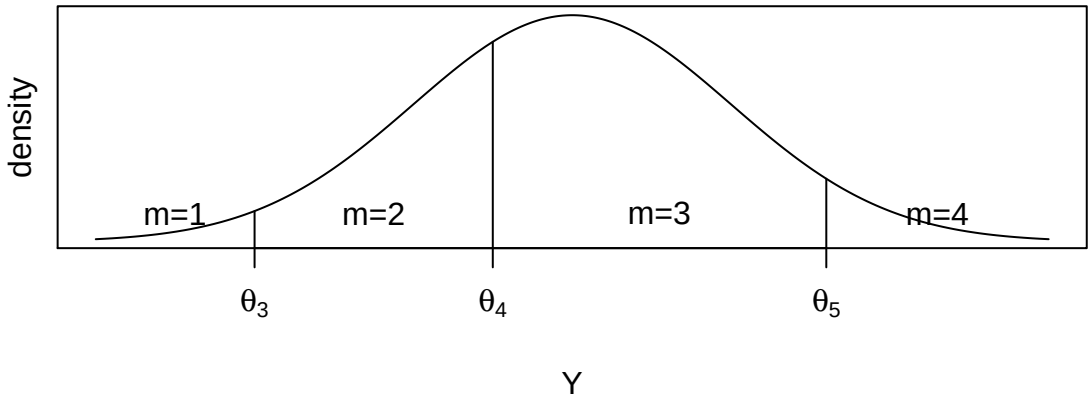
$$m_i = \begin{cases} 1 & \text{if} \theta_2 < y_i < \theta_3 \\ 2 & \text{if} \theta_3 < y_i < \theta_4 \\ \vdots & \\ t & \text{if} \theta_{t+1} < y_i < \theta_{t+2} \end{cases}$$

where, θ are the cut points used to discretize $\boldsymbol{Y}$ into $\boldsymbol{M}$, and %i% is the number of bins. For notational convenience, $\theta_2 \equiv -\infty$ and $\theta_{t+2} \equiv \infty$.[5]

To give a concrete example, the following figure shows the density of $\boldsymbol{Y}$ when the cuts points are, for this example, $\theta = (-\infty, -2, -0.5, 1.6, \infty)$. In this example, any value of $-2 < y_i < -0.5$ would have $m_i = 2$.

[4] For a more complete treatment of the polyserial correlation, see Cox, N. R., "Estimation of the Correlation between a Continuous and a Discrete Variable" *Biometrics*, **50** (March), 171-187, 1974.

[5] The indexing is somewhat odd to be consistent with Cox (1974). Nevertheless, this treatment does not use the Cox definition of θ_0, θ_1 or θ_2 which are either not estimated (as is the case for θ_0, and θ_1) or are reappopriated (as is the case for θ_2). Cox calls the correlation coefficient θ_2 while this document uses ρ and uses θ_2 to store $-\infty$ as a convenience so that the vector $\boldsymbol{\theta}$ includes the (infinite) bounds as well as the interior points.



Notice that μ_y is not identifed (or is irrelevant) because, for any $a \in \mathbb{R}$, setting $\tilde{\mu}_y = \mu_y + a$ and $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + a$ lead to exactly the same values of $\mathbf{M}$ and so one of the two must be arbitrarily assigned. A convenient decision is to decide $\mu_y \equiv 0$. A similar argument holds for σ_y so that $\sigma_y \equiv 1$.

For $\boldsymbol{X}$, Cox (1974) observes that the MLE mean and standard deviation of $\boldsymbol{X}$ are simply the average and (population) standard deviation of the data and do not depend on the other parameters.[6] This can be taken advantage of by defining z to be the standardized score of x so that $z \equiv \frac{x-\bar{x}}{\hat{\sigma}_x}$.

Combining these simplifications, the probability of any given x_i, m_i pair is

$$\Pr\left(\rho = r, \theta; Z = z_i, M = m_i\right) = \phi(z_i) \int_{\theta_{m_i+1}}^{\theta_{m_i+2}} f(y|Z = z, \rho = r) dy$$

where $\Pr\left(\rho = r, \theta; Z = z, M = m\right)$ is the probability of the event $\rho = r$ and the cuts points are θ, given the data z and m; $\phi(\cdot)$ is the standard normal; and $f(y|z, \rho)$ is the distribution of y conditional on z and ρ. Because y and z are jointly normally distributed (by assumption)

$$f(y|Z = z, \rho = r) = N\left(\mu_y + \frac{\sigma_y}{\sigma_z} r(z - \mu_z), (1 - r^2){\sigma_y}^2\right)$$

because both $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are standard normals

$$f(y|Z = z, \rho = r) = N\left(r \cdot z, (1 - r^2)\right)$$

now, define $w \equiv \frac{y - r \cdot z}{\sqrt{1-r^2}}$ and w has a standard normal distribution. Plugging this in

$$\Pr\left(\rho = r, \theta; Z = z_i, M = m_i\right) = \phi(z_i) \left[\Phi\left(\frac{\theta_{m_i+2} - r \cdot z_i}{\sqrt{1 - r^2}}\right) - \Phi\left(\frac{\theta_{m_i+1} - r \cdot z_i}{\sqrt{1 - r^2}}\right)\right]$$

Where $\Phi(\cdot)$ is the standard normal cumulative density function. Using the above probability function as an objective, the log-likelihood is then maximized.

$$\ell(\rho, \theta; z, m) = \sum_i w_i \ln\left[\Pr\left(\rho = r, \theta; Z = z_i, M = m_i\right)\right]$$

where w_i is the weight of the i^{th} members of the vectors $\boldsymbol{Z}$ and $\boldsymbol{Y}$. For the unweighted case, all of the weights are set to one.

The value of the nuisance paramter $\boldsymbol{\theta}$ is chosen to be $\Phi^{-1}(n/N)$ where n is the number of values to the left of the cut point (θ_i value) and N is the number of data points overall.

[6] The population standard devaition is used because it is the MLE for the standard deviation. Notice that, while the sample variance is an unbiased estimator of the variance and the population variance is not an unbaised estimator of the variance, they are very similar and the variance is also a nuisance parameter, not a parameter of interest when finding the correlation.

Computation of polyserial correlation

The derivatives of ℓ can be written down but are not readily computed. When the `ML` argumet is set to `FALSE` (the default), a one dimensional optimization of ρ is calculated using `stats::optimize` at the values of $\boldsymbol{\theta}$ from the previous paragraph. When the `ML` argument is set to `TRUE`, a multi-dimensional optimization is done for ρ and $\boldsymbol{\theta}$ using `minqa::bobyqa`. See the *wCorr Arguments* vignette for a comparison of these two methods.

Because the numerical optimization is not perfect when the correlation is in a boundary condition ($\rho \in \{-1, 1\}$), a check for perfect correlation is performed before the above optimization by simply examining if the values of $\boldsymbol{X}$ and $\boldsymbol{M}$ have agreeing order (or opposite but agreeing order) and then the MLE correlation of 1 (or -1) is returned.

Methodology for polychoric correlation with and without weights

Similar to the polyserial correlation, the polychoric correlation is a simple case of two continuous variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ that have a bivariate normal distribution. In the case of the polyserial correlation the continuous (latent) variable $\boldsymbol{Y}$ was observed as a discretized variable $\boldsymbol{M}$. For the polychoric correlation, this is again true but now the continuous (latent) variable $\boldsymbol{X}$ is observed as a discrete variable $\boldsymbol{P}$ according to

$$p_i = \begin{cases} 1 & \text{if} \theta'_2 < x_i < \theta'_3 \\ 2 & \text{if} \theta'_3 < x_i < \theta'_4 \\ \vdots & \\ t & \text{if} \theta'_{t+1} < x_i < \theta'_{t+2} \end{cases}$$

where $\boldsymbol{\theta}$ remains the cut points for the distibution defining the transformation of $\boldsymbol{Y}$ to $\boldsymbol{M}$ and $\boldsymbol{\theta}'$ is the cut points for the transformation from $\boldsymbol{X}$ to $\boldsymbol{P}$. Similar to $\boldsymbol{\theta}$, $\boldsymbol{\theta}'$ has $\theta'_2 \equiv -\infty$ and $\theta'_{t+2} \equiv \infty$.

As in the polyserial correlation, μ_y is not identifed (or is irrelevant) because, for any $a \in \mathbb{R}$, setting $\tilde{\mu}_y = \mu_y + a$ and $\tilde{\boldsymbol{\theta}}' = \boldsymbol{\theta}' + a$ lead to exactly the same values of $\mathbf{M}$ and so one of the two must be arbitrarily assigned. The same is true for μ_x. A convenient decision is to decide $\mu_y = \mu_x \equiv 0$. A similar argument holds for σ_y and σ_x so that $\sigma_y = \sigma_x \equiv 1$

Then the probability of any given m_i, p_i pair is

$$\Pr\left(\rho = r, \theta, \theta'; P = p_i, M = m_i\right) = \int_{\theta'_{p_i+1}}^{\theta_{p_i+2}} \int_{\theta_{m_i+1}}^{\theta_{m_i+2}} f(x, y|\rho = r) dy dx$$

where ρ is the correlation coefficient.

Using this function as an objective, the log-likelihood is then maximized.

$$\ell(\rho, \theta, \theta'; p, m) = \sum_i w_i \ln\left[\Pr\left(\rho = r, \theta, \theta'; P = p_i, M = m_i\right)\right]$$

This is the weighted log-likelihood function. For the unweighted cause all of the weights are set to one.

Computation of polychoric correlation

This again mirrors the treatment of the polyserial. The derivatives of ℓ can be written down but are not readily computed. When the `ML` argument is set to `FALSE` (the default), a one dimensional optimization of ρ is calculated using `stats::optimize`. When the `ML` argument is set to `TRUE` a multi-dimensional optimization is done for ρ, $\boldsymbol{\theta}$, and $\boldsymbol{\theta}'$ using `minqa::bobyqa`. See the *wCorr Arguments* vignette for a comparison of these two methods.

Because the optimization is not perfect when the correlation is in a boundary condition ($\rho \in \{-1, 1\}$), a check for perfect correlation is performed before the above optimization by simply examining if the values of $\boldsymbol{P}$ and $\boldsymbol{M}$ have a Goodman-Kruskal correlation coefficient of -1 or 1. When this is the case, the MLE of -1 or 1, respectively, is returned.

Correctness of the estimating methods

It is easy to prove the consistency of the $\boldsymbol{\theta}$ for the polyserial correlation and $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ for the polychoric correlation using the non-ML case. Similarly, for ρ, because it is an MLE that can be obtained by taking a derivative and setting it equal to zero, the results are asymtotically unbiased and obtain the Cramer-Rao lower bound.

This does not speak to the small sample properties of these correlation coefficients. Previous work has described their properties by simulation; and that tradition is continued below.

Simulation study of unweighted correlations

In what follows, when the exact method of selecting a parameter (such as n) is not noted in the above descriptions it is described as part of each simulation.

A simulation is run several times (the exact number of times will be stated for each simulation). For each iteration, the following procedure is used:

- select a true correlation coefficient ρ;
- select the number of observations (n);
- generate $\boldsymbol{X}$ and $\boldsymbol{Y}$ to be bivariate normally distributed using a pseudo-Random Number Generator (RNG);
- using a pseudo-RNG, select the the number of bins for $\boldsymbol{M}$ and $\boldsymbol{P}$ (t and t') independantly from the set {2, 3, 4, 5};
- select the bin boundaries for $\boldsymbol{M}$ and $\boldsymbol{P}$ ($\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$) by sorting the results of $(t-1)$ and $(t'-1)$ draws, respectively, from a normal distribution using a pseudo-RNG;
- confirm that at least 2 levels of each of $\boldsymbol{M}$ and $\boldsymbol{P}$ are occupied (if not, retrun to generating $\boldsymbol{X}$ and $\boldsymbol{Y}$); and
- calculate and record relevant statistics.

Bias, and RMSE of the unweighted correlations

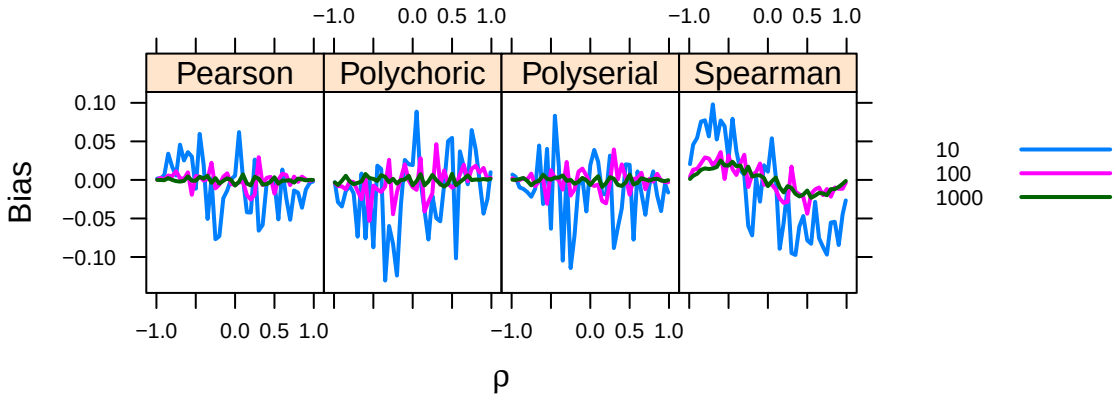
This sections shows the bias of the correlations as a function of the true correlation coefficient, ρ. To that end, a simulation was done at each level of the cartesian product of $\rho \in (-0.99, -0.95, -0.90, -0.85, ..., 0.95, 0.99)$, and $n \in \{10, 100, 1000\}$. For precision, each level of ρ and n was run fifty times. The bias is the mean difference between the true correlation coefficient (ρ_i) and estimate correlation coefficient (r). The RMSE is the square root of the mean squared error.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_i (r_i - \rho_i)^2}$$

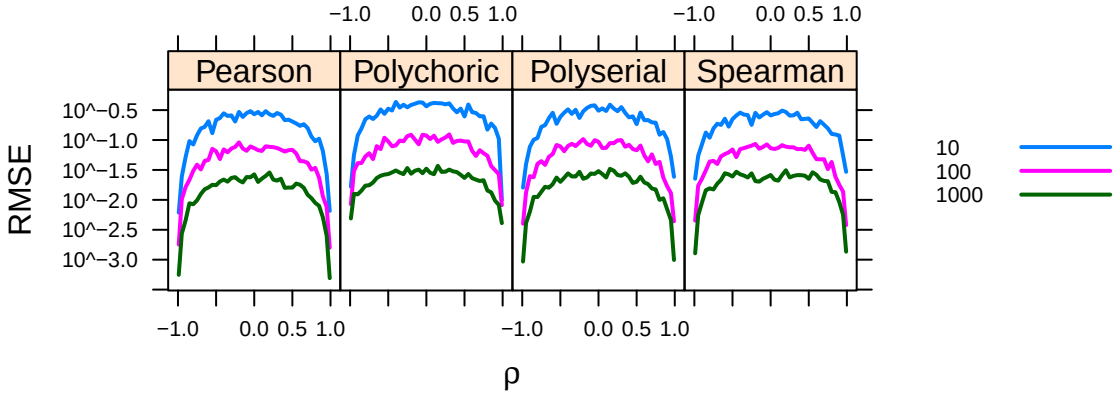
And the bias is given by

$$bias = \frac{1}{n}\sum_i (r_i - \rho_i)$$

This plot shows the bias as a function of the true correlation ρ. Only the Spearman correlation shows a clear trend with a positive bias below 0 (negative correlation) and a negative bias above 0 (positive correlation). The other estimators show no clear bias.

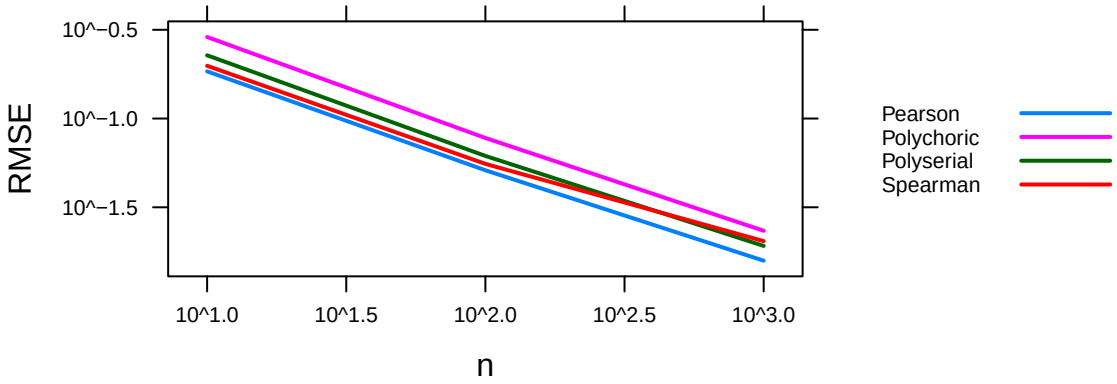


This plot shows the RMSE as a function of ρ. All of the correlation coefficients have a uniform RMSE as a function of ρ near $\rho = 0$ that decreases near $|\rho| = 1$. All plots also show a decrease in RMSE as n increases. This plot shows that there is no appreciable RMSE differences as a functions of ρ. In addition, it show that our attention to the MLE correlation of -1 or 1 at edge cases did not make the RMSE much worse in the neighborhood of the edges ($|\rho| \sim 1$).



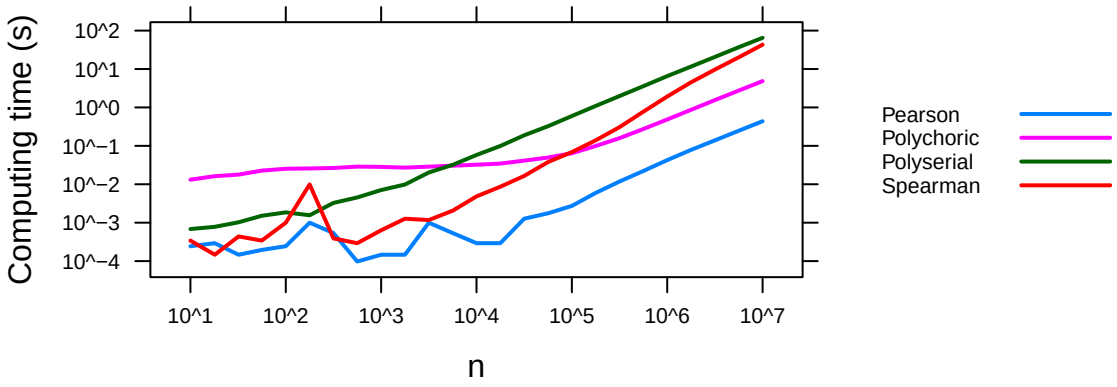
Consistency of the correlations

This plot shows the RMSE as a function of n. The plot shows a slope of about $-\frac{1}{2}$, which is consistent with the expected first order convergence for each correlation coefficient under the assumptions of this simulation.



Computing Time

This plot shows the mean time (in seconds) to compute a single correlation coefficient as a function of ρ by n size. The plot shows linearly rising computation times with slopes of about one. This is consistent with a linear computation cost. Using Big O notation, the computation cost is, in the range shown, O(n).



Simulation study of weighted correlations

When complex sampling (other than simple random sampling with replacement) is used, unweighted correlations may or may not be consistent. In this section the consistency of the weighted coefficients is examined.

When generating simulated data, decisinos about the generating functions have to be made. These decisions affect how the results are interpreted. For the weighted case, if these decisions lead to something about the higher weight cases being different from the lower weight cases then the test will be more informative about the role of weights. Thus, while it is not reasonable to always assume that there is a difference between the high and low weight cases, the assumption (used in the simulations below) that there is an association between weight and the correlaiton serves as a more robust test of the methods in this package.

Results of weighted correlation simulations

Simulations are carried out in the same fashion as previously described but include a few extra steps to accomodate weights. The following changes were made:

- weights are assigned according to $w_i = (x - y)^2 + 1$, and the probability of inclusion in the sample was then $\Pr_i = \frac{1}{w_i}$.
- For each unit, a uniformly distributed random number was drawn. When that value was less than the probability of inclusion ($\Pr_i$) the unit was included.

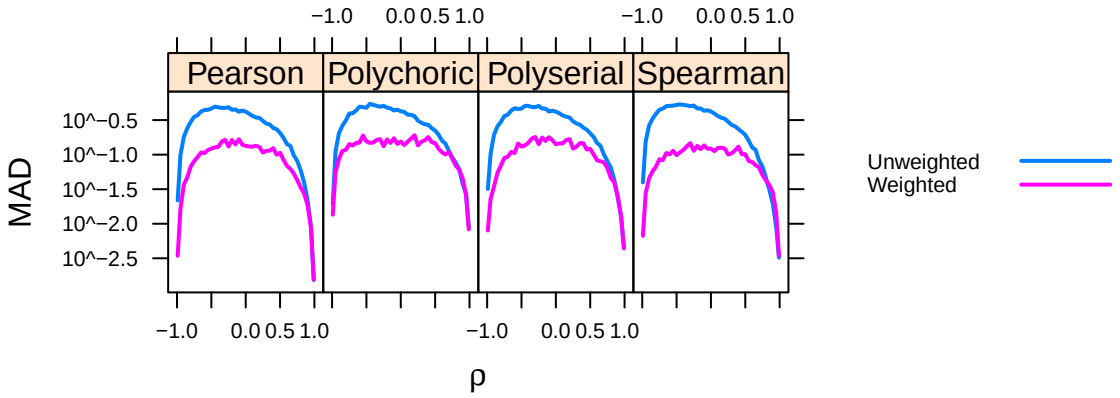
Units were generated until n units were in the sample.

Two simulations were run. The first shows the mean absolute deviation (MAD)

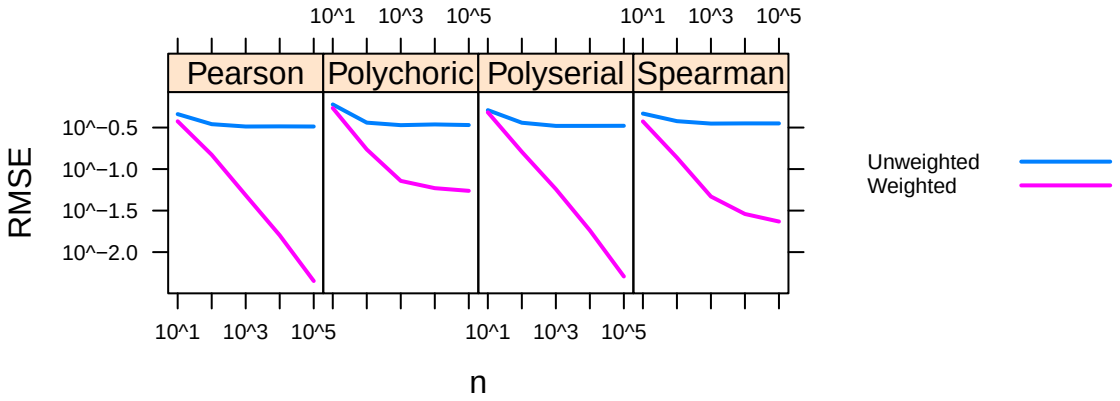
$$MAD = \frac{1}{n} \sum_i |r_i - \rho_i|$$

as a function of ρ and was run for $n = 100$ and $\rho \in (-0.99, -0.95, -0.90, -0.85, ..., 0.95, 0.99)$, with 100 iterations run for each value of ρ.

The following plot shows the MAD for the weighted and unweighted results as a function of ρ when $n = 100$. This shows that for values of ρ near zero, under our simulation assumptions, the weighted correlation performs better than (has lowe MAD than) the unweighted correlation for all correlation coefficients. Over the entire range, the difference between the two is never such that the unweighted has a lower MAD.



The second simulation used the same values of ρ and used $n \in \{10, 100, 1000, 10000, 100000\}$ and shows how RMSE and sample size are related. In particular, it shows first order convergence of the weighted Pearson and polyserial correlation coefficient. The polychoric and Spearman are not consistent and have minimum RMSE values indicating that some residual error always exists–under the assumptions of our simulation. In all cases the RMSE is lower for the weighted than the unweighted. Again, the fact that the simulations show that the unweighted correlation coefficient (and some of the weighted correlation coefficients) is not consistent is not meant to imply that it will always be that way–only that this is possible for these coefficients to not be consistent.


Pearson
Polychoric
Polyserial
Spearman
RMSE
10^−0.5
10^−1.0
10^−1.5
10^−2.0
10^1
10^3
10^5
n
Unweighted
Weighted